

FILE COPY

UNLIMITED

DR 10 71 12

2



AD-A205 879

RSRE -
MEMORANDUM No. 4133

ROYAL SIGNALS & RADAR ESTABLISHMENT

ROUTE DATA REPRESENTATION FOR
ATC SIMULATION EXPERIMENTS

Author: W Finch

RSRE MEMORANDUM No. 4133

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

DTIC
ELECTE
MAR 29 1989
S H D

UNCLASSIFIED

UNLIMITED

0032203

CONDITIONS OF RELEASE

BR-109148

.....

U

COPYRIGHT (c)
1988
CONTROLLER
HMSO LONDON

.....

Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 4133

TITLE: ROUTE DATA REPRESENTATION FOR ATC SIMULATION EXPERIMENTS
AUTHOR: W FINCH
DATE: October 25, 1988

Abstract

The route data representation described was written to support research by the TCSDG (Terminal Control Systems Development Group) at RSRE Malvern into Air Traffic Management and Control, sponsored by the UK NATS (National Air Traffic Services). The representation is used in conjunction with an Air Traffic Simulator and an Air Traffic Management System. It was written in CORAL 66 to run on a VAX 11/780 computer.

CARDPB Ref : 2.2.1.2
RSRE Task : 636
CAA Sponsors : D PE(LD)
CAA Liaison Officer : D Dawson (DPA1)

This memorandum is for advance information. It expresses the views of the author and is not necessarily to be regarded as a final or official statement by the Civil Aviation Authority nor the Procurement Executive, Ministry of Defence.

Copyright

©

Controller HMSO London

1988

- 1 INTRODUCTION.
- 2 REQUIREMENTS.
 - 2.1 General Requirements.
 - 2.2 Special Requirements.
- 3 THE PHYSICAL MODEL.
 - 3.1 Routes.
 - 3.1.1 General Description.
 - 3.1.2 Branches.
 - 3.1.3 Common Segments.
 - 3.2 Beacons and Airports.
 - 3.3 Stacks.
 - 3.4 Sectors.
- 4 THE ROUTE DATA INPUT FILE.
 - 4.1 Beacon and Airport Specifications.
 - 4.2 Sector Specifications.
 - 4.3 Common Route Segment Specifications.
 - 4.4 Route Specifications.
 - 4.5 Waypoint Lists.
 - 4.6 Waypoint Qualifiers.

Association For
NILE TRUST
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654

- 5 THE ROUTE COMPILER.
- 6 PROGRAM DATA STRUCTURES.
 - 6.1 Route Pointer Table.
 - 6.2 Way Point Table.
 - 6.3 Beacon Table.
 - 6.4 Stack Table.
 - 6.5 Sector Table.
 - 6.6 Branch Table.
- 7 APPLICATIONS.
- 8 CONCLUSIONS.
- 9 ACKNOWLEDGEMENTS.
- 10 REFERENCES.
- 11 APPENDICES A AND B.

ROUTE DATA REPRESENTATION FOR AIR TRAFFIC CONTROL SIMULATION EXPERIMENTS

1 INTRODUCTION.

The use of simulated air traffic environments is central to the development of many air traffic control systems. An important element of these systems is the data structure used to describe air routes and their properties. This paper is concerned with an air route data structure developed as part of an Air Traffic Control research programme carried out by the Terminal Control Systems Development Group (TCSDG) at RSRE Malvern. The research programme is aimed at formulating and evaluating new ideas and procedures for a future air traffic control system. The research is sponsored by the UK National Air Traffic Services (NATS). The software developed as part of the research programme was written in CORAL 66 and runs on a VAX 11/780 computer.

The route structure representation was designed to model a large area around south east England. This area contains several major airports and many minor airports and is extremely complicated in ATC terms due to a mix of climbing, descending and overflying aircraft. The route structure described has been used in a real-time simulation of a skeleton air traffic control system using computer assistance for arrivals management [1]. This system is referred to as the Air Traffic Management (ATM) system. Associated with this ATM system was an air traffic simulator [2] which also makes extensive use of the route structure. This air traffic simulator is referred to as the Simulator. Within these systems it is necessary to display route data and also operate on it by :-

- (a) Navigation (by the Simulator)
- (b) Trajectory prediction (by the ATM system)
- (c) Reading the ATC constraints and data as applicable to each way point on the route (both systems)

2 REQUIREMENTS.

2.1 General Requirements.

The main criterion when designing the representation was that it should adequately model both the geography in terms of airports, navigational aids etc and also the ATC route dependant environment such as sectorisation, height, speed constraints and holding fixes.

It was also important, because of its intended use in support of a research programme, that the representation could be easily understood and quickly modified to meet the needs of each experiment. These requirements together with the fact that it would be used in many modules, on and off line, led to the adoption of the following strategies:-

- (1) The modularity of the data structure definition and size

using a macro file.

- (2) The use of a self contained data area at run time using a global section.
- (3) The plain text description of the route data input file.

2.2 Special Requirements.

A route data structure for use in an experimental environment needs to be planned to take account of the range of possible uses to which it might be put. A well structured design will make changes and extensions easier and will allow maximum use of common access procedures with a consequential reduction in the need for new software to cope with every new use or changed data content.

In this instance the route data structure must include two basic elements. These are the geographical features regarding the navigational beacons and their positions and secondly the air traffic control features such as airport facilities, holding stacks and sectorisation. The ATM and the air traffic Simulator systems both need to know the exact path to be taken by aircraft and also need to be aware of any ATC constraints that might apply at any particular way point on the route being flown.

The main processes which access the routes, including the off line utilities are :-

- (1) The aircraft prediction module in the ATM. This is a set of procedures for predicting future aircraft trajectories given an aircraft's current route and position. These procedures need to know in which direction the aircraft is going, and how and where it will turn next etc etc.
- (2) The process that draws the synthetic radar displays. This process uses the list of beacons and their associated attributes to provide the controller with a synthetic radar display showing beacons and way points as symbols.
- (3) The navigational part of the air traffic Simulator to fly aircraft along the routes.
- (4) The off line process that generates the traffic samples used in the various experiments [3]. This process generates aircraft traffic at specified densities and mixes on the routes required.
- (5) The off-line process that plots the route data structure showing also the ATC constraints. This utility produces a hard copy in colour, an example of the output is shown in fig 2. See also [4]. The output from this process has proved to be very useful for visualising the layout of routes when modifying the route data input file.

For ease of operation the software was written as a stand alone module writing data into a dedicated data area at run time. Data structure definitions were located in one file. Any process

requiring to use it would first define a data area and then call a procedure that would read the input file and set this data area accordingly.

The main operations that are required to be performed on the data structure are as follows:-

- (1) To read the beacon list table to plot the beacon positions on a radar display and optionally give them names and symbols. Also for airports to draw the runways with the correct alignment.
- (2) To scan the list of routes to check whether a particular route is valid given an entry and exit point. (Note that it is assumed that all routes are uniquely specified by an entry/exit point combination.
- (3) To scan a list of waypoints for a particular route to find the next waypoint and any associated ATC constraint.
- (4) To scan a list of waypoints for a particular route to find a specified ATC constraint on the route. eg the holding fix for arriving traffic.
- (5) To scan a list of waypoints for a particular route to check if a particular beacon or waypoint is used.
- (6) To scan a list of waypoints for a particular route from a specified point either forwards or backwards to calculate the position and time taken to perform a particular manoeuvre.
- (7) To scan the complete route structure and ATC constraints to draw any selected route or group of routes as required.
- (8) To scan a list of waypoints on a particular route for Branch qualifiers, to enable a temporary copy of the waypoint list for the route to be generated using the branch option names specified.

3 THE PHYSICAL MODEL.

As mentioned in the Requirements section the representation needed to model the whole environment in which Air Traffic Control normally operates. This section briefly describes the various features of this environment and indicates their interrelationships.

3.1 Routes.

3.1.1 General Description.

A route describes a path for an aircraft from entry fix to exit point and is defined in terms of a series of way points. These waypoints are defined in terms of beacons, are not necessarily co-located with a beacon, and may have qualifiers which define how the way point is to be used in air traffic control terms.

In the structure described here there can be different paths from entry to exit because easterly and westerly and also low and high performance routes can be defined. Easterly and westerly routes are required for inbound and outbound aircraft as the runway landing direction at the airport may be changed dependant upon the prevailing wind direction or other factors. The two landing directions may require differing way point lists particularly in the vicinity of the airport. High and low performance routes are required mainly for outbound aircraft. They are used particularly on or near crossing routes. Aircraft that can achieve higher rates of climb that enable them to clear other traffic are given what is termed a high performance route. Aircraft unable to achieve the rate of climb required are given a low performance route. The Simulator generates aircraft positions based normally on the assumption that the aircraft is on the designated route. However this is not always the case.

3.1.2 Branches.

Two circumstances have been identified which indicate that it would be useful to be able to set an aircraft onto a alternative but well defined path for part of its route during run time. Firstly aircraft on their approach may or may not be required to hold. If an aircraft is not required to hold then it may be turned away from a route defined through the stack point to a more direct path to touch down. Secondly if a simulation of MLS (Microwave Landing System) curved path approaches was taking place then each aircraft would need to be set, by the controller, onto that path through the MLS space which best provided a safe and timely arrival at the runway. In both of these cases knowledge of the selected alternative path is essential so that the prediction algorithm within the Air Traffic Management System can continue to function correctly.

To cover these cases the concept of a branch has been introduced. A branch defines named alternative paths, each defined as a series of way points. Where a branch is specified on a route, only the first option at each branch point is flown by aircraft by default. To select the other options, a temporary copy of the route with the new way points is set into the dynamic data area, based on a list of the branch name options specified. The aircraft may then set to fly this new copy of the route. A sample branch specification is shown in Appendix B page 3.

3.1.3 Common Segments.

Many routes in the airspace have portions that use the same waypoints and have identical ATC constraints. It is useful to be able to identify these as common segments in the route input file. In addition, defining them in only one place rather than duplicating information reduces the possibility of consistency errors being introduced when routes are changed.

3.1.4 Beacons and Airports.

Beacons are instances of physical navigation facilities. They are normally VOR (VHF Omni-directional Radio) or DME (Distance Measuring Equipment). They can also, however be the position markers of an Airport ILS (Instrument Landing System). They are defined by an identifier or name with positional data.

3.1.5 Stacks.

Stacks or Holding Areas are volumes, within the airspace being modelled, in which aircraft are held until they can be accepted for approach and touchdown. They are defined in terms of a stack point, which is a waypoint with an appropriate qualifier, and are linked to a list of physical attributes such as sense and orientation of the holding pattern and also the upper and lower flight levels allowed.

3.1.6 Sectors.

Sectors are volumes of airspace defined in order to bound and define a controller's area of responsibility. In this Route Data Representation the way points are used, by way of a qualifier, to define points at which a sector change occurs and to indicate a new sector.

4 THE ROUTE DATA INPUT FILE.

A route consists of a list of way points. These waypoints are defined in terms of beacons, and may have qualifiers which define

how the way point is to be used in air traffic control terms. Because it is desirable that the contents of the route data input file be as comprehensible as possible to programmers and more importantly to non-programmers (for example, air traffic controllers), the data is expressed in textual form using a data description language. This syntax-driven approach makes the route compiling program more complex, but offers a significant benefit in terms both of data checking and readability.

There now follows a brief description of the language used; a full syntax is given in Appendix A together with a sample input shown in Appendix B.

The input data can be written in a free format except for certain requirements such as the need to define beacon names and common segment lists before they are used in a route specification. The input can include bracketed comments to amplify the meaning. (This facility was useful and has also enabled us to remove temporarily any part of the route structure and then insert it again quickly and accurately).

4.1 Beacons and Airport Specifications.

Beacon specifications comprise a name followed by positional and attribute data. Positional data can be specified as a latitude and longitude or as an X and Y displacement, in nautical miles from a fixed beacon designated as the system origin. If a latitude and longitude is used it is converted by the route compiler into X and Y values. This conversion is done using gnomonic projection based on the tangent plane through one particular beacon. The fixed beacon chosen for our experiments was the Bovingdon (BNN) VOR/DME beacon. Attributes specify whether the beacon represents an airport or not (and if so give the runway orientation and length of the extended runway centre line to be drawn on the radar display) and they also define the symbol type to be used on the radar display or the plotted map and whether the beacon name is to be displayed or not. See later for a complete list of way point attributes.

4.2 Sector Specifications.

ATC sector names can be specified along with the name or symbol required to be shown on the radar display and other air traffic control displays. For a sector sample specification see Appendix B. The syntax is sector name followed by symbol usually a single character terminated with a semicolon. Sector names are quoted with the 'HO=name' waypoint qualifier (described later).

4.3 Common Route Segment Definitions.

Common segments are defined by giving a name (preferably chosen to be meaningful, e.g. R1N-IN for an inbound route along airway Red One North) followed by a list of waypoints (For an example see Appendix B Page 3). Common segments are used in the route specifications (See next) by quoting the common segment name instead

of a way point.

4.4 Route Specifications.

Route specifications are divided into three types: arrivals, departures and overflights. All three types contain a single letter to indicate route type. For arrivals and departures this is followed by the airport name and two lists of waypoints and/or common segments with one list for an easterly landing direction and the other for westerly. The language demands that both lists are given, though of course they can be identical. A further option can divide these two lists into high and low performance routes by using language words LOW and HIGH. High and Low performance in this context refers to an aircraft's climb performance and is usually only required for departing aircraft. For overflights the route type is followed by a single list of waypoints and/or common segments.

The possibility of allowing alternative paths within a route was introduced for later experiments mainly concerned with approach control. Here a special way point qualifier called a 'Branch' has been introduced. This is discussed later.

4.5 Waypoint Lists.

Waypoints are points marking the route and can correspond to navigational beacons, but can also be offset from such beacons. Waypoints that are co-located with a beacon are specified simply by writing the beacon name. Offset waypoints can be specified in one of two ways. Firstly a waypoint qualifier 'DIS=' can be given together with a beacon name and a track distance. It is assumed that the named beacon is also a waypoint in its own right along the route. The waypoint position is defined as the specified number of miles before the beacon along the track line between the previous waypoint and the named beacon. The second method uses a waypoint qualifier '[R=n1 DME=n2]' to specify a point relative to a beacon at the given DME distance 'n2' along the specified radial 'n1'. The named beacon need not necessarily be on the route. As already indicated the branch qualifier allows alternative route legs to be specified.

4.6 Waypoint Qualifiers.

The concept of waypoint qualifiers has already been introduced by example. Any waypoint can be qualified with a variety of qualifiers and these are now described. In the following list the default qualifier values are given in brackets. If several waypoint qualifiers are required for a single waypoint they are separated by commas. No limit is placed on the number of such qualifiers for a particular waypoint. For a complete syntax of the input file see Appendices A and B. A list of valid waypoint qualifiers now follows.

- /APP - waypoint marks the start of the Approach Control phase

- /APT - waypoint is an airport (Set by default on first waypoint for departures and last waypoint for arrivals)

- /BRANCH [Branch Specification]
 - waypoint marking the start of a branch. ie the start of a branch node. Branch leg names and way point lists follow in the [Branch Specification]. For the complete syntax see Appendix A.

- /DIS = n - offset waypoint at n miles before the qualified beacon along the route (default zero)

- /ENDSID - waypoint marks the end of a Standard Instrument Departure route SID

- /ENT - the waypoint at which an aircraft on the route enters the simulation, normally the start of the route. (Set by default on the first waypoint for arrivals and overflights)

- /HMIN = n
/HMAX = n - specifies minimum/maximum flight levels at the waypoints (defaults zero for HMIN and 999 for HMAX)

- /HO = <Sector Name>
 - waypoint at which hand-over to/from another sector takes place (set by default on first waypoint for arrivals and first and last waypoints for overflights). A sector name is required to be quoted as listed in the sector table

- /IAP - waypoint is an Inbound Assembly Point. The precise meaning is defined by the ATM system.

- /NOOVERTAKE - No overtaking allowed beyond this waypoint. The precise meaning is defined by the ATM system.

- /NWP - waypoint not to be shown on the route plotting map

- /OVERFLY - aircraft is to overfly the waypoint and not to anticipate the turn to the next Way Point (default is no overfly)
- /RWYCL - put on waypoints close to the Airport which lie on the extended runway centre line. Used to ensure correct ILS capture in the Simulator
- /STK [sense,orientation Base-level:Ceiling-level]
 - waypoint has a stack pattern defined by the specified sense and orientation. Base-level marks the lower Flight Level while Ceiling-level marks the upper flight level (default sense is right-hand and orientation the aircraft's heading on reaching the holding fix)
- /SPDMAX = n - specifies maximum speed of n knots to apply at this waypoint (default 999)
- /STOP - waypoint at which an aircraft on the route leaves the simulation, normally the last waypoint (set by default on the last waypoint)
- /[R=n1 DME=n2]
 - offset waypoint at range n2 miles from the qualified beacon along the radial given by n1 (default is DME = 0)

5 THE ROUTE COMPILER.

The route compiler is a piece of software which converts the route input data file into the program data structures. It is invoked by a procedure call and is invoked by all processes listed in section 2 which use the Route Data Input File as source data. It is built as a stand alone module and is syntax-driven using a recursive-descent parsing method. It is relatively easy to implement syntax changes should this prove necessary.

Beacon data is read directly into the beacon table. Common segment definitions are stored in local or temporary data structures. Common segment data is held in the Way Point Table during the processing of the input file. The data area used is released once the main route data has been read. Route descriptors and waypoint lists are built directly from the route specifications. Default qualifier data is set where data is not specified.

The program uses hashing techniques to improve the efficiency of beacon name searches. (On average the number of comparisons per name access for 150 names is two). The program takes two parameters. The first one is an array of airport landing

directions. This is used to determine whether the easterly or westerly route specified in the route specification is the one to be setup in the data structure. The second parameter is used to return a textual error message if required. This is discussed later.

When reading way point qualifiers recursive procedures are used as it may be necessary to read further qualifiers after reading a branch qualifier. A branch itself reads way points and way point qualifiers. To simplify the programming overhead concerned with branches only two levels of branching have been allowed. Explicit branching within branches is not allowed but a common segment containing a branch may be called within a branch called in the main route specification area.

Any errors with the syntax or data space are reported as errors giving the type of error and the last item read in from the data file. The line number in the input data file containing the error is also quoted.

6 PROGRAM DATA STRUCTURES

The main data structures and their interrelationships are shown in Figure 1. Routes are stored as a set of route descriptors indexed by a route number. At runtime either Easterly or Westerly routes (High and Low performance) are loaded for each airport based upon data read from an environment data file regarding the wind speed and direction etc. It is not possible to change the direction of routes during a run. Each descriptor describes certain route properties as shown in the Route Pointer Table and contains pointers to the waypoints comprising the low and high performance routes. The waypoint table contains a set of records, one for each waypoint. Waypoints for a route are chained together in a list. Each waypoint is associated with one beacon by a reference to the appropriate entry in the beacon table. References to other tables such as the Stack Table, Sector Table and Branch Table are made as required. The waypoint properties field contains most of the waypoint qualifier data. Various bits are set in the field for each qualifier as required.

Common segments have no realisation at run time; they use temporary data structures as the information is not required once the main route specification has been read and the way point list assembled.

Any size of route structure can be loaded up to a fixed limit which was designed to be large enough for anticipated requirements. Extra routes can be added dynamically if required. This feature has in fact been used with the introduction of branch elements. If an alternative route needs to be selected at a branch node, then a copy of the route with only the required way points is generated and copied into the dynamic area. When the aircraft exits the system the dynamic route is released and put onto a free list.

A more detailed description of the route data structures now follows.

6.1 Route Pointer Table.

This table holds the basic data concerning a route. From it enough information may be obtained to pick up the required list of way points for a particular route.
The fields are:-

- (a) Route Type. The type of route either Arrival, Departure or Overflight.
- (b) Route Frequency. This field refers to the initial simulated R/T frequency that the aircraft is assigned to on entry to the system.
- (c) Route Airport. This field refers to the airport used by the route. eg usually Heathrow, Gatwick, Luton or Stanstead.
- (d) Route Pointer High. This field points to the first way point in the Way Point Table for the route used by aircraft able to use the high performance category.
- (e) Route Pointer Low. This field points to the first way point in the Way Point Table for the route used by aircraft unable to use the high performance category

6.2 Way Point Table.

This table holds the way point lists for each route specified. References are made to the other tables in the data structure such as the Beacon Table, Stack Table, Sector Table and Branch Table as required.
The fields are:-

- (a) WP Beacon Reference. This field supplies the reference to the Beacon Table.
- (b) WP Distance from Beacon. This holds the distance in nautical miles from the beacon specified in (a).
- (c) WP Position X and Y. These fields hold the position of the way point in nautical miles from Bovingdon (BNN) VOR/DME beacon.
- (d) WP Property. This field is a word with bits set marking the various way point qualifiers. If the way point is a stack then a reference is also made to the Stack Table.
- (e) WP Maximum Height. This field holds the maximum allowed flight level at the way point if the appropriate bit has been set in WP Property.
- (f) WP Minimum Height. This field holds the minimum flight level at the way point if the appropriate bit has been set in WP Property.

- (g) WP Maximum Speed. This field holds the maximum speed allowed at the way point if the appropriate bit has been set in WP Property.
- (h) WP Sector. This field holds the reference to the Sector Table if the Hand Over qualifier is used on the way point.
- (i) WP Branch. This field refers to the Branch Table if the branch qualifier is used on the way point.
- (j) WP Next. This field points to the next way point on the route. A value of -1 indicates that the end of the route has been reached.

6.3 Beacon Table.

This table holds the list of beacons read in from the beacon specifications from the first part of the input file.
The fields are:-

- (a) Name. This is a reference to the beacon name eg BCN for Brecon VOR/DME beacon.
- (b) Position X. X distance in nautical miles from Bovingdon (BNN) beacon.
- (c) Position Y. Y distance in nautical miles from Bovingdon (BNN) beacon.
- (d) Attributes. This field holds the information about how the beacon is to be drawn on the radar display. eg the type of symbol to be used and if the name is to be show etc.

6.4 Stack Table.

This table holds the holding stack data.
The fields are:-

- (a) Stack Beacon. A reference to the Beacon Table giving the name of the stack.
- (b) Stack Base Height. The lower flight level to be used for the stack.
- (c) Stack Ceiling. The upper flight level to be used for the stack.
- (d) Stack Orientation. The orientation in degrees of the inbound leg of the holding pattern. ie the heading of the aircraft when returning to the holding fix.
- (e) Stack Sense. The sense of the turn (right or left) of aircraft flying over the stack beacon about to fly the holding pattern.

6.5 Sector Table.

This table holds the sectorisation data.
The fields are:-

- (a) Sector Name. A reference to the sector name as specified in the input data file.
- (b) Sector Designator. The symbol or string to be associated with the sector name and shown on the radar display.
- (c) Sector Frequency. The simulated R/T frequency to be assigned to the Sector.

6.6 Branch Table.

This table holds the branch data. Each branch leg is listed in the table.
The fields are:-

- (a) Branch Name. This field holds the Branch Leg name as read from the input file.
- (b) Branch Start Pointer. This field holds the reference in way point table for the start of the branch leg.
- (b) Branch End Pointer. This field holds the reference in way point table for the end of the branch leg.
- (b) Branch Next Leg. This field refers to the next branch leg in the Branch Table. A value of -1 indicates the end of the branch section.

7 APPLICATIONS.

As already described the route structure has been used both in an Air Traffic Simulator and an Air Traffic Management system. An aircraft prediction module has also been written using the structure. This module is used within the Management system to predict accurately the position of an aircraft in three dimensions both in forwards and backwards prediction modes.

An important off line programme using the route data structure is the Traffic Sample Generator [3]. This module generates a flight plan script for both the ATM and Simulator systems based on a specification file giving traffic density loadings and aircraft type mixes for each route.

Associated with both the Simulator and Management systems,

described above, modules driving displays use the data structure to generate synthetic radar displays. These displays show beacon positions and other data along with the current aircraft positions in a plaque to the controllers working on the system.

The route data input file has also proved to be very useful as a discussion document both before and after experiments. As an aid to realisation, an offline program has been written to produce a hard copy of the route structure by way of a plotter. This program uses the routes data file as input and any route or collection of routes may be drawn in colour and can also show beacons and waypoints as required. The ATC constraints at any waypoint on the route can be shown if required. By this method complex route interactions can be drawn graphically and this has proved to be very useful for removing conflicts between the various routes, for example, between arrivals and departures.

8 CONCLUSIONS.

The route structure has been used in many TCSDG simulation experiments over the last 4 years. The structure has proved to be very flexible in operation and has adapted very well to the simulations required. Indeed the flexibility of the system was severely tested when the TCSDG route structure was changed in 1986 from a system using high level holding with 4 airports to one using only low level holding with up to 10 airports. This change was made with only minor problems. Other changes to the route structure have also been made on a day to day basis quite quickly and easily.

On the debit side, the common route segment feature although good in concept has proved to have two problems associated with it. Firstly it removes a little of the transparency of the input file and secondly it has been observed that although aircraft may pass the same waypoints on their route, differing ATC constraints such as allowed heights and speeds are often imposed at these waypoints.

9 ACKNOWLEDGEMENTS.

The author would like to thank all persons who have contributed to the design of the TCSDG route data structure over the time of the project especially A C F Tyler, S A N Magill, A J Budd and E T Wilkinson.

10 REFERENCES.

- [1] S A N Magill, A C F Tyler, E T Wilkinson and W Finch. Computer based tools for assisting Air Traffic Controllers with arrivals flow management. RSRE Report 88001. 1988.
- [2] S A N Magill, W Finch, A C F Tyler and E T Wilkinson. A real-time simulator which provides an environment for

air traffic control experiments. Proceedings SCS European Simulation Conference, Antwerp. September 1986.

[3] S A N Magill, G Ord and K A L Burek. A traffic sample generator for Air Traffic Control simulation experiments. RSRE Memo 3818. 1988.

[4] A users guide to Rplot (Internal RSRE Note) A J Budd 1987.

11 APPENDICES.

APPENDIX A
SYNTAX FOR SPECIFICATION FILE

Note the syntax definitions that follow use Bachus-Naur (BNF) notation and the following symbols and definitions are used:-

::=	'is defined to be'
	OR
()	Comment statements inclosed.
{ }	Brackets enclose anything that may appear zero or more times.
CAPITALS	Terminal function or symbol.
< >	Non terminal symbols.

A.1 BEACON SPECIFICATIONS

The syntax for specifying beacons is as follows:-

```
BEACONS
<Beacon Specification>
{<Beacon Specification>}
ENDBEACONS
```

Beacon Specification ::=
 <Beacon Name> <Coordinate Spec> <Attributes> ; (Semicolon Terminator)

<Beacon Name> is a string of alpha and numeric characters beginning with an alpha character. It may contain underscore characters but not spaces.

<Coordinate Spec> ::= <Coordinate Type> <Coordinates> |
 <VOR range/DME Spec>

```
<Attributes> ::=
    APT <ILS Heading for airport on westerly operations> |
    VM1 (Beacon to be displayed as video map symbol '1') |
    VM2 (Beacon to be displayed as video map symbol '2') |
    SN (Show name on video map)
```

A.2 SECTOR SPECIFICATIONS

The syntax for specifying sectors is as follows:-

```
SECTORS
<Sector Specification>
{<Sector Specification>}
ENDSECTORS
```

Sector Specification ::=
 <Sector Name> <Sector Symbol> ; (Semicolon Terminator)

<Sector Name> is a string of alpha and numeric characters beginning with an alpha character. It may contain underscore characters but not spaces.

<Sector Symbol> is an alpha or numeric character.

A.3 ROUTE SPECIFICATIONS

Route Specification ::=
 {<Common Segment Spec>} <Route Spec>

<Common Segment Spec> ::=
 COMMON_SEGMENTS
 <Common Segment Data>
 {<Common Segment Data>}
 END_COMMON_SEGMENT

<Common Segment Data> ::= <Common Segment Name>:<Common Segment Way Point List>;

<Common Segment Name> is a string of alpha and numeric characters beginning with an alpha character. It may contain underscore characters but not spaces.

<Common Segment Way Point List> ::= <Waypoint> {<Waypoint>}

<Waypoint> ::= <Way Point Name> {<Way Point Qualifier List>}

<Way Point Name> is a string of alpha and numeric characters beginning with an alpha character. It may contain underscore characters but not spaces.

<Way Point Qualifier List> ::= / <Qualifier> { , <Qualifier> }

<Route Spec> ::= <Arrival Route> |
 <Departure Route> |
 <Overflight Route>

<Arrival Route> ::=
 A <Airport Name> - <Landing Direction> {<Performance Category>}
 <Route Way Point List>
 {<Performance Category> <Route Way Point List>}
 <Landing Direction> {<Performance Category>}
 <Route Way Point List>
 {<Performance Category> <Route Way Point List>} ;

<Departure Route> ::=
 D <Airport Name> - <Landing Direction> {<Performance Category>}
 <Route Way Point List>
 {<Performance Category> <Route Way Point List>}
 <Landing Direction> {<Performance Category>}
 <Route Way Point List>
 {<Performance Category> <Route Way Point List>} ;

<Airport Name> is a string of alpha and numeric characters beginning with an alpha character. It may contain underscore characters but not spaces.

<Overflight Route> ::=
 O : <Route Way Point List> ;

<Route Way Point List> ::= <Waypoint> {<Waypoint>} |
 <Waypoint> {<Common Segment Name>} |

```

    <Common Segment Name> {<Waypoint>} |
    <Common Segment Name> {<Common Segment Name>}

```

```

<qualifier> ::= APP | (Start of Approach Control)
               APT | (Airport)
               BRANCH <Branch Spec> | (Branch or alternative route begins
                                         here. Lists of alternative legs as
                                         sets of waypoints follow after each
                                         branch leg name)
               DIS=n | (Offset Waypoint n miles before designated waypoint)
               ENDSID | (End of Standard Instrument Departure routine for
                        departing aircraft)
               ENT | (Entry waypoint)
               HMIN=n | (Minimum height constraint n=Flight Level)
               HMAX=n | (Maximum height constraint n=Flight Level)
               HO=SECTOR-NAME-STRING | (Sector handover point)
               IAP | (Inbound Assembly Point)
               NOOVERTAKE | (No overtaking allowed at this way point)
               NWP | (Way point not to be shown on the route plotting
                     Map)
               OVERFLY | (Way point to be overflown. ie the turn to the
                          next way point may not be anticipated)
               RWYCL | (Way point on the extended runway centre line)
               STK=[<sense,Orientation=n LOW-FL:HIGH-FL>] | (Holding stack
                     with sense of L=left R=right. Orientation is in
                     degrees. Lower and upper allowed flight levels
                     are specified in LOW-FL and HIGH-FL)
               SPDMAX=n | (Speed limit point of n Knots)
               STOP | (End of simulation waypoint)
               [R=n1 DME=n2] (Offset way point on radial n1 degrees at
                             range n2 miles)

```

```

<Branch Spec> ::= {<Leg name string> = <Route Way Point List> .
                  {<Leg name string> = <Route Way Point List>}}

```

<Leg name string> is a string of alpha and numeric characters beginning with an alpha character. It may contain underscore characters but not spaces.

APPENDIX B

SAMPLE BEACON AND ROUTE SPECIFICATIONS

B.1 NOTE

In the following paragraphs text enclosed in round brackets is treated as a comment by the software. Note that the following text is for information and example only and has been severely edited. Consequently the text would not necessarily compile as many beacon, sector names, and common segments have been excluded for simplicity. It is shown here to enable the reader to form an idea of what the structure of the input file could look like.

B.2 BEACON SPECIFICATIONS

(Common Control System - Beeker - Routes)

BEACONS

(Airports)

EGBB	LL 52:27:00N 01:45:00W APT [ILS= 326];	(Birmingham)
EGGW	LL 51:53:00N 00:22:00W APT [ILS= 255];	(Luton)
EGHI	LL 50:57:00N 01:21:00W APT [ILS= 199];	(Southampton)
EGKB	LL 51:19:00N 00:02:00E APT [ILS= 205];	(Biggin Hill)
EGKK	LL 51:09:00N 00:11:00W APT [ILS= 258];	(Gatwick)
EGLL	LL 51:28:00N 00:27:00W APT [ILS= 270 ECL= 10];	(Heathrow)
EGNX	LL 52:50:00N 01:19:00W APT [ILS= 268];	(East Midlands)
EGSS	LL 51:53:00N 00:14:00E APT [ILS= 223];	(Stansted)

(VOR DME Beacons)

ABB	LL 50:08:00N 01:51:00E VM1 SN;	(Abbeville)
BIG	LL 51:19:49N 00:02:11E VM1 SN;	(Biggin)
BKY	LL 51:59:44N 00:03:59E VM1 SN;	(Barkway)
BNN	LL 51:43:30N 00:32:51W VM1 SN;	(Bovingdon)
BCN	LL 51:43:26N 03:15:42W VM1 SN;	(Brecon)
BNE	LL 50:37:00N 01:54:00E VM1 SN;	(Boulogne)
BRU	LL 50:52:00N 04:20:00E VM1 SN;	(Arrival Entry Beacon)
BUR	LL 51:31:02N 00:40:09W VM1 SN;	(Burnham)
CFD	LL 52:04:26N 00:36:35W VM1 SN;	(Cranfield)
CLN	LL 51:51:00N 01:09:00E VM1 SN;	(Clacton)
COA	LL 51:22:00N 03:20:00E VM1 SN;	(Costa)

(NDB Beacons)

BPK	LL 51:45:00N 00:06:00W VM1 SN;	(Brookman's park)
CON	LL 53:11:48N 02:11:35W VM1 SN;	(Congleton)
EPM	LL 51:19:00N 00:22:00W VM1;	(Epsom)

SAMPLE BEACON AND ROUTE SPECIFICATIONS

Page B-2

GAB LL 51:59:23N 02:04:38E VM2 SN; (Gabbard)
 GWC LL 50:51:26N 00:45:06W VM1 SN; (Goodwood)
 GY LL 51:07:49N 00:18:52W VM1; (Gatwick)

(Reporting points)

ALY LL 51:33:00N 01:59:54W VM1; (Abeam Lyneham)
 AMM LL 51:50:00N 04:00:00W VM1 SN; (Ammanford)
 BLF LL 53:08:00N 03:17:00E VM2 SN; (Bluefir)
 CLY LL 51:40:00N 01:07:00W VM1; (Cowley)
 LNX XY -210.129 -96.020 VM2;

(Arrival stacks)

(Heathrow)

MKE BNN 319R 19DME VM1 SN; (Milton keynes)
 BKA BNN 261R 9DME VM1 SN; (Booka)
 CVY BIG 058R 22DME VM1 SN; (Canvy)
 LAM BPK 130R 12DME VM1 SN; (Lambourne)
 OXS BIG 280R 10DME VM1 SN; (Oxshott)

(Gatwick)

MYW SFD 285R 17DME VM1 SN; (Mayfield West)
 CMA MAY 110R 23DME VM1 SN; (Camba)

ENDBEACONS

SECTORS

HURN H; (inbound and outbound)
 IRISH_SEA R; (inbound and outbound)
 NORTH N; (inbound and outbound)
 BRISTOL B; (inbound)
 CLACTON C; (inbound)
 LYDD L; (inbound)
 SEAFORD W; (inbound)
 DOVER D; (outbound)

END_SECTORS

(TCSDG Route specifications)

(A = Arrival)
 (D = Departure)
 (O = Overflight)
 (E = Easterly route)
 (W = Westerly route)
 (LOW = Low performance route)
 (HIGH = High performance route)

(ROUTES)

COMMON_SEGMENTS

(Arrival common segments)

(Departure common segments)
(LL WESTERLY)

(Overflight Common Segments)
(NOTE Only High Level Routes are included here)

```

A [EGLL] - W R1N_IN LL_LAM_IN
           - E R1N_IN LL_LAM_IN;

A [EGLL] - W R1S_IN LL_LAM_IN
           - E R1S_IN LL_LAM_IN;

A [EGLL] - W B1_IN LL_LAM_IN
           - E B1_IN LL_LAM_IN;

A [EGLL] - W B29_IN LSD2_LL_LAM_IN
           - E B29_IN LSD2_LL_LAM_IN;

```

A [EGKK] - W R1N DVR IN KK SE IN

- E R1N_DVR_IN KK_SE_IN;
A [EGKK] - W R1S_DVR_IN KK_SE_IN
- E R1S_DVR_IN KK_SE_IN;
A [EGKK] - W G1E_IN LYD3_DIS_15/HMIN=150,HMAX=150 LYD3/HO=CMA_SECTOR
KK_SE_IN
- E G1E_IN LYD3_DIS_15/HMIN=150,HMAX=150 LYD3/HO=CMA_SECTOR
KK_SE_IN;

(DEPARTURES)

D [EGLL] - W HIGH: LL10LMM/SPDMAX=250
EPM/[R=324 DME=7],HMAX=070,SPDMAX=250
EPM/OVERFLY,HMIN=040,HMAX=070,SPDMAX=250,END_SID
DET/[R=277 DME=32],SPDMAX=250,HMAX=170
DET/HMAX=170,SPDMAX=250,HO=DOVER
NF/HMAX=170 KOK/HO=POST_DVR
- W LOW: LL10LMM/SPDMAX=250
EPM/[R=324 DME=7],HMAX=060,SPDMAX=250
EPM/OVERFLY,HMIN=030,HMAX=060,SPDMAX=250,END_SID
DET/[R=277 DME=32],SPDMAX=250,HMAX=170
DET/HMAX=170,SPDMAX=250,HO=DOVER
NF/HMAX=170 KOK/HO=POST_DVR
- E HIGH: LL28RMM/SPDMAX=250
DET/[R=288 DME=23],HMIN=050,HMAX=070,SPDMAX=250,END_SID
DET/HMAX=170,SPDMAX=250,HO=DOVER
NF/HMAX=170 KOK/HO=POST_DVR
- E LOW: LL28RMM/SPDMAX=250
DET/[R=288 DME=23],HMAX=040,SPDMAX=250,END_SID
DET/HMAX=170,SPDMAX=250,HO=DOVER
NF/HMAX=170 KOK/HO=POST_DVR;

(OVERFLIGHTS)

O: OVF_DCS_CPT BCN/DIS=40,HO=BRISTOL BCN/HO=POST_STU;
O: OVF_DCS_CPT SAM/DIS=15,HO=HURN SAM ORT/STOP,HO=POST_HURN DIN;
O: DCS/HO=PRE NORTH CON2/HO=NORTH CON HON BPK/HO=LAM_SECTOR
CLN/DIS=15,HO=CLACTON CLN RFN/HO=POST_CLN SPY;

(ENDROUTES)

(END OF FILE)

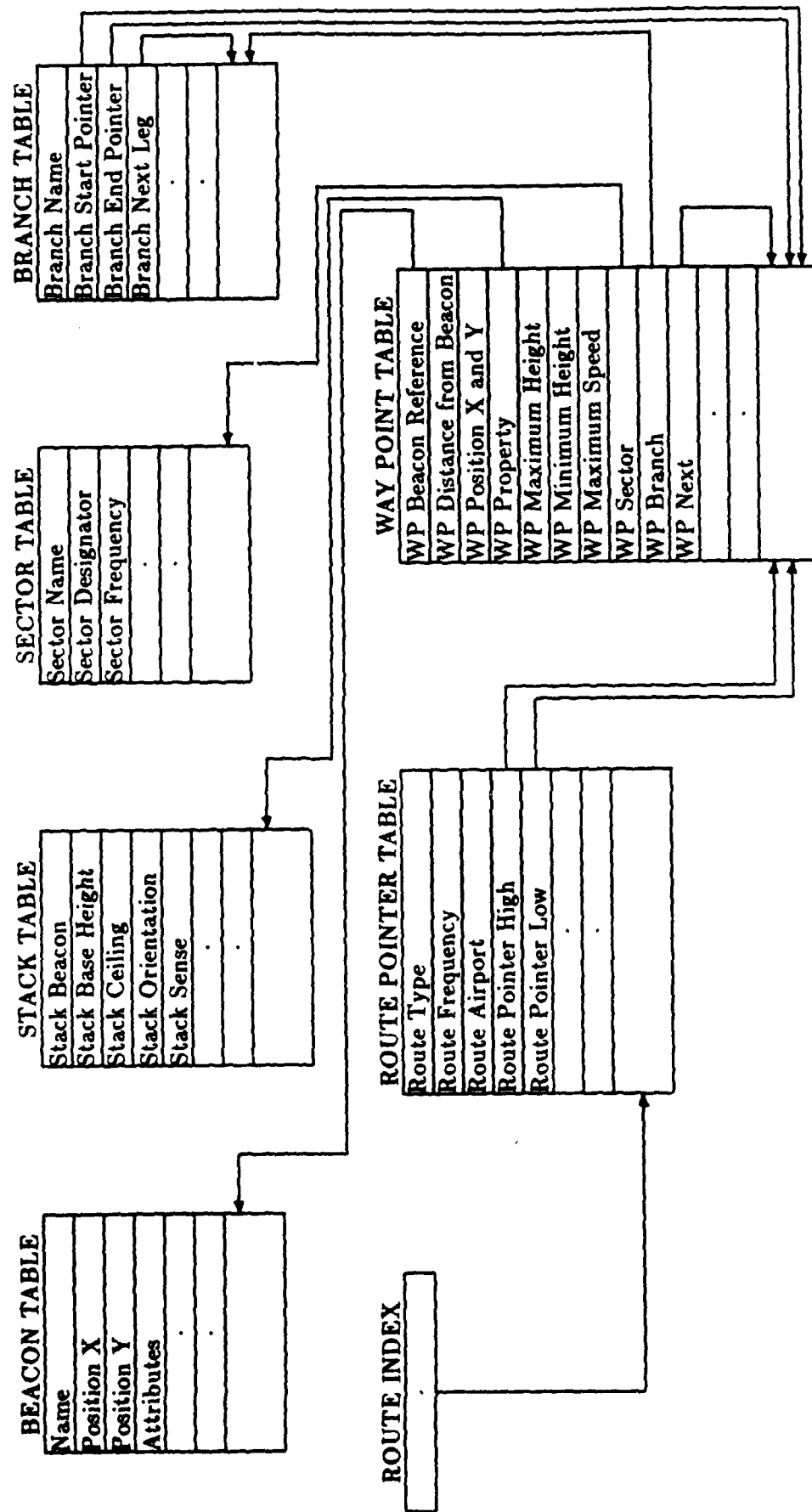


Figure 1 Route Data Structures

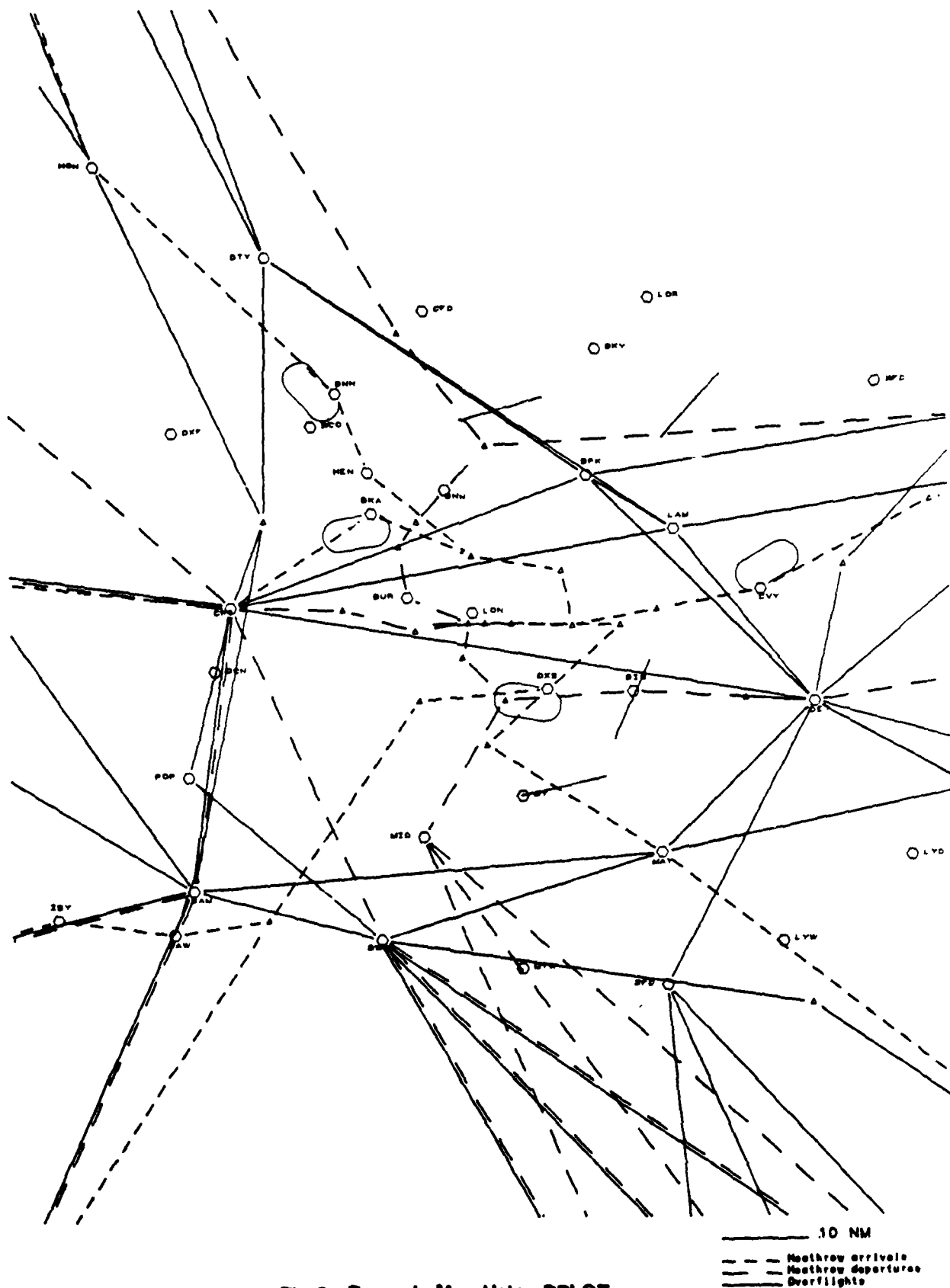


Fig 2 Example Map Using RPLOT

DOCUMENT CONTROL SHEET

Overall security classification of sheet UNCLASSIFIED

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference Memo 4133	3. Agency Reference	4. Report Security U/C Classification	
5. Originator's Code (if known) 7764000	6. Originator (Corporate Author) Name and Location ROYAL SIGNALS & RADAR ESTABLISHMENT ST ANDREWS ROAD, GREAT MALVERN, WORCESTERSHIRE WR14 3PS			
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location			
7. Title Route data Representation for ATC simulation experiments				
7a. Title in Foreign Language (in the case of translations)				
7b. Presented at (for conference papers) Title, place and date of conference				
8. Author 1 Surname, initials Finch W	9(a) Author 2	9(b) Authors 3,4...	10. Date 1988.10	pp. ref. 27
11. Contract Number	12. Period	13. Project	14. Other Reference	
15. Distribution statement Unlimited				
Descriptors (or keywords) continue on separate piece of paper				
Abstract The route data representation described was written to support research by the TCSDG (Terminal Control Systems Development Group) at RSRE Malvern into Air Traffic Management and Control, sponsored by the UK NATS (National Air Traffic Services). The representation is used in conjunction with an Air Traffic Simulator and an Air Traffic Management System. It was written in CORAL 66 to run on a VAX 11/780 computer. — (5.2.1 2.1.1) —				